

1.

-Tính kế thừa cho phép các lớp được xây dựng dựa trên các lớp đã có.

-Tính kế thừa dùng để biểu diễn mối quan hệ đặc biệt hóa- tổng quát hóa giữa các lớp. Trong đó lớp dẫn xuất kế thừa lớp cơ sở.

-Lớp dẫn xuất kế thừa tất cả các thành phần của lớp cơ sở (cả thuộc tính và phương thức) kể cả các thành phần lớp cơ sở đó được thừa kế.

Ứng dụng của tính kế thừa:

-Phát triển chương trình

-Sửa đổi, nâng cấp, bổ sung chương trình

-Thiết kế bài toán từ khái quát đến cụ thể, từ chung đến riêng

-Thiết kế bài toán chung và bài toán bộ phận.

Vd:

```
Class HìnhHoc {
```

```
Public:
```

```
Void tinhdientich();
```

```
};
```

```
Class HìnhTron: public HìnhHoc{
```

```
Public:
```

```
Void tinhchuvi();
```

```
}
```

Trong ví dụ trên thì lớp con HìnhTron kế thừa cách tính diện tích từ lớp cha và có thể khai báo riêng một lớp chu vi cho chính nó.

*Đoạn chương trình trên xảy ra lỗi vì chưa khai báo constructor không có tham số cho lớp A. Để sửa lỗi trên ta cần khai báo constructor không có tham số cho lớp A như sau:

```
Class A{  
A(){};  
A(int i);  
};
```

2.

Khai báo và cài đặt lớp phân số:

```
#include <iostream>
```

```
using namespace std;
```

```
class PhanSo {  
private:  
int tuSo, mauSo;  
public:  
PhanSo(){tuSo=0; mauSo=1;};  
PhanSo(int t, int m){tuSo=t; mauSo=m;};  
PhanSo(int t){tuSo=t; mauSo=1;};  
PhanSo operator+ (const PhanSo& p);  
void Xuat();  
};  
void PhanSo::Xuat(){  
cout<<this->tuSo<<"/"<<this->mauSo<<endl;  
}  
PhanSo PhanSo::operator+(const PhanSo& p){
```

```
PhanSo kq;  
kq.tuSo= this->tuSo*p.mauSo + this->mauSo*p.tuSo;  
kq.mauSo=this->mauSo*p.mauSo;  
return kq;  
}
```

3.

